

On Comparative Study of Deterministic Linear Consensus-based Algorithms for Distributed Summing

Martin Kenyeres
Institute of Informatics
Slovak Academy of Sciences
 Bratislava, Slovakia
 martin.kenyeres@savba.sk

Jozef Kenyeres
Sipwise GmbH
 Brunn am Gebirge, Austria
 jkenyeres@sipwise.com

Abstract—Application of data aggregation mechanisms is supposed to ensure high confidence of measurements and low energy demands in wireless sensor networks. Therefore, many modern applications utilize distributed algorithms for aggregate function estimation in order to minimize negative factors affecting the operation of the wireless sensor networks. This paper is concerned with deterministic linear consensus-based algorithms for distributed summing or more specifically, a comparative study of five frequently applied algorithms from this algorithm category over random graphs and random geometric graphs. The selected algorithms are examined using various methodologies and metrics.

Keywords—distributed computing; distributed summing; wireless sensor networks

I. INTRODUCTION

In many WSN¹-based applications (e.g., event detection, target tracking, decision making, etc.), one of the most fundamental issues is to ensure the sensor readings with high confidence and low energy requirements in spite of imprecision of the sensor nodes [1-2]. WSNs are formed by spatially distributed sensor nodes for collecting and processing information about the observed physical quantity [3]. These sensor nodes communicate with each other (or with a base station) via wireless channels in order to make a meaningful decision about the observed phenomenon [4-5]. The sensor readings are affected by several factors (e.g., strong variations of pressure, various noises, duplicated data, node failures, etc.) worsening QoS² of the executed applications [1]. Therefore, many WSN-based applications are equipped with data aggregation algorithms for minimizing the impacts of the negative effects, whereby QoS is enhanced, and the energy consumption can be optimized [1]. Xiao et al. define two categories of data aggregation algorithms in [6]. In the centralized data aggregation schemes, the measured data is collected and processed by a data fusion center. These mechanisms often require routing mechanisms at

each sensor node, which is not an appropriate solution, especially for mobile or battery constrained networks. The modern applications are usually equipped with distributed data aggregation schemes with no central fusion center and absents any global information about the network topology. As shown in the literature [7-10], consensus-based algorithms for data aggregation (among others deterministic linear consensus-based distributed summing algorithms (DLCDSAs) with time-invariant and symmetric weights) find usage in various WSN-based applications due to their lower computational, communication, memory, and energy demands.

In this paper, we analyze five frequently applied DLCDSAs (namely, the Maximum Degree weights, the Metropolis-Hastings, the Local Degree weights, the Best Constant weights, and Optimized Convex weights algorithms) over 1000 random graphs (RG) formed by 50 nodes with a varying probability of the edge formation and 100 random geometric graphs (RGGs) formed by 200 nodes with either dense or sparse connectivity. We use two research methodologies and evaluate the performance of the analyzed DLCDSAs using the following metrics: the mean square error over the iterations (MSE(k)) and the convergence rate expressed as the number of the iterations necessary for all the sensor nodes to achieve the consensus.

The second section of this paper is focused on modeling WSNs and chosen DLCDSAs using tools defined in the graphs theory and the linear algebra. The third section is concerned with the applied research methodologies, and the fourth one consists of the experimentally obtained results and a discussion about them.

II. THEORETICAL BACKGROUND

A. Model of DLCDSAs and WSNs

We model WSNs as indirect finite graphs G determined by two sets \mathbf{V} and \mathbf{E} ($G = (\mathbf{V}, \mathbf{E})$) [11]. The set \mathbf{V} consists of all the vertices in a graph representing the particular nodes present in the inspected network. Each vertex is allocated a unique identification number for easy identification, i.e., $\mathbf{V} = \{v_1, v_2, \dots, v_n\}$, where n is the order of the inspected graph (i.e., $n = |\mathbf{V}|$). The set

This work was supported by the VEGA agency under the contract No. 2/0155/19 and by the Slovak Research and Development Agency under the contract No. APVV-17-0116.

¹Wireless sensor network

²Quality of Service

\mathbf{E} contains all the edges in the inspected graph, which represent the direct connection between two vertices (labeled as (v_i, v_j) or alternatively, e_{ij}).

DLCDSAs can be in general modeled as the difference equation defined as follows [12]:

$$\mathbf{x}(k+1) = \mathbf{W} \times \mathbf{x}(k) \quad (1)$$

Here, \mathbf{W} is the weight matrix determined by used DL-CDSA. Its elements affect several aspects of distributed summing such as the convergence rate, the robustness, the initial configuration, etc. [11]. Furthermore, $\mathbf{x}(k)$ is a column vector variant over the iterations gathering the inner states of all the sensor nodes at the k^{th} iteration. Subsequently, each sensor is able to estimate the sum of all inner states as follows [13]:

$$x_i^{\text{sum}}(k) = n \cdot x_i(k), \forall v_i \in \mathbf{V} \quad (2)$$

In DLCDSAs, the inner state at each sensor node asymptotically converges to the sum calculated from the initial inner states (i.e., $\mathbf{x}(k=0)$) of all the sensor node in an inspected network, which can be expressed as follows [14]:

$$\lim_{k \rightarrow \infty} \mathbf{x}(k) = \lim_{k \rightarrow \infty} \mathbf{W}^k \times \mathbf{x}(0) = \frac{1}{n} \cdot \mathbf{1} \times \mathbf{1}^T \times \mathbf{x}(0) \quad (3)$$

Here, $\mathbf{1}$ is an all-ones column vector, having "1" in each position [15]. The existence of this limit is crucial for the proper functioning of DLCDSAs and is ensured if these following convergence conditions are met [12]:

$$\mathbf{1}^T \times \mathbf{W} = \mathbf{1}^T \quad (4)$$

$$\mathbf{W} \times \mathbf{1} = \mathbf{1} \quad (5)$$

$$\rho(\mathbf{W} - \frac{1}{n} \cdot \mathbf{1} \times \mathbf{1}^T) < 1 \quad (6)$$

Here, $\rho(\cdot)$ is the spectral radius of the analyzed matrix/vector defined as follows [16]:

$$\rho(\cdot) = \max_i \{ |\lambda_i(\cdot)| : i = 1, 2, \dots, n \} \quad (7)$$

Thus, it represents the largest Laplacian eigenvalue in the absolute value.

B. Analyzed DLCDSAs

In this section, we introduce examined DLCDSAs and provide their mathematical definitions. The first algorithm of our interest is the Maximum Degree weights algorithm (MD), which is described by the Perron matrix as follows [11]:

$$[W]_{ij}^{\text{MD}} = \begin{cases} \frac{1}{\Delta}, & \text{if } e_{ij} \in \mathbf{E} \\ 1 - d_i \cdot \frac{1}{\Delta}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Here, Δ is the maximum degree of the analyzed graph, and d_i is the degree of the vertex v_i (i.e., the number of its neighbors) [11].

The next algorithms of our interest are the Metropolis-Hastings (MH) and the Local Degree (LD) weights algorithms, characterizable by non-uniform edge weights. They require only locally obtainable information for their proper initial configuration and

therefore find wide application in WSNs. Their weight matrices are defined as follows [11-12]:

$$[W]_{ij}^{\text{MH}} = \begin{cases} \frac{1}{\max\{d_i, d_j\} + 1}, & \text{if } e_{ij} \in \mathbf{E} \\ 1 - \sum_{i \neq j} [W]_{ij}^{\text{MH}}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

$$[W]_{ij}^{\text{LD}} = \begin{cases} \frac{1}{\max\{d_i, d_j\}}, & \text{if } e_{ij} \in \mathbf{E} \\ 1 - \sum_{i \neq j} [W]_{ij}^{\text{LD}}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

The Best Constant weights algorithm (BC) is considered to be the fastest among the algorithms describable by the Perron matrix. It requires the exact value of the second smallest and the largest Laplacian eigenvalues ($\lambda_{n-1}(\mathbf{L})$ and $\lambda_1(\mathbf{L})$) for its proper initial configuration. Its weight matrix is defined as follows [12]:

$$[W]_{ij}^{\text{BC}} = \begin{cases} \frac{2}{\lambda_1(\mathbf{L}) + \lambda_{n-1}(\mathbf{L})}, & \text{if } e_{ij} \in \mathbf{E} \\ 1 - d_i \cdot \frac{2}{\lambda_1(\mathbf{L}) + \lambda_{n-1}(\mathbf{L})}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

The last analyzed algorithm is the Optimized Convex weights algorithm (OW), which is designated in such a way that the fast linear averaging problem³ is expressed as the spectral radius minimization problem as:

$$\begin{aligned} & \text{minimize} && \rho(\mathbf{W} - \frac{1}{n} \cdot \mathbf{1} \times \mathbf{1}^T) \\ & \text{subject to} && \mathbf{L} \in \mathbf{S}, \mathbf{1}^T \times \mathbf{W} = \mathbf{1}^T, \mathbf{W} \times \mathbf{1} = \mathbf{1} \end{aligned} \quad (12)$$

Here, the parameter \mathbf{S} represents the Laplacian matrix sparsity pattern limits.

III. RESEARCH METHODOLOGY

In this section, we introduce the applied research methodologies and the used metrics for performance evaluation in our analysis. Our experiments are executed in two types of graphs:

- *random graphs* [17] - are formed by 50 vertices, and the existence of an edge is determined by the probability p , which takes these four values: 3.5%, 5%, 10%, and 15%. The graphs are invariant over the iterations, and for each probability p , 1000 unique graphs are generated.
- *random geometric graphs* [17] - are formed by 200 vertices. We generate two sets of these graphs - densely and sparsely connected and show their representatives in Fig. 1. The graphs are invariant over the iterations again, and each set consists of 100 unique graphs.

Furthermore, we assume that each sensor node is allocated a random scalar value of the standard Gaussian distribution representing local measurement and determining the initial state of the corresponding sensor node (the inner states are independent and identically distributed), i.e.:

$$x_i(0) \sim N(0, 1), \text{ for } \forall v_i \in \mathbf{V} \quad (13)$$

In this paper, we carry out these two scenarios:

³originally, it was demonstrated on distributed averaging

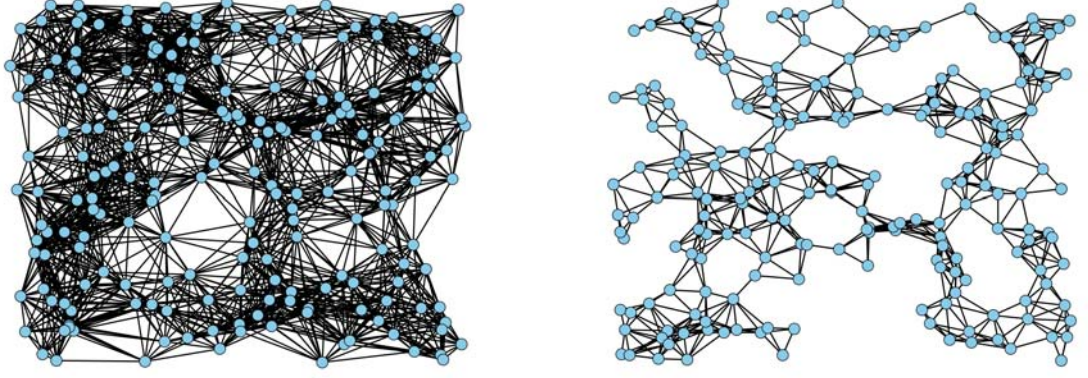


Fig. 1. Representatives of densely (figured left) and sparsely (figure right) connected random geometric graphs

- *the execution of selected DLCDsAs is not bounded* - we choose seven iterations⁴ at which the algorithms are analyzed - MSE averaged over all 1000 RGs⁵/100 RGGs⁶ at these iterations
- *the execution of selected DLCDsAs is bounded by a stopping criterion* - the algorithm execution is bounded by the stopping criterion defined as follows [11]:

$$| \max\{\mathbf{x}(k)\} - \min\{\mathbf{x}(k)\} | < \mathcal{P} \quad (14)$$

Here, \mathcal{P} determines the precision of the final estimates, however, higher precision results in a lower convergence rate. In our experiments, we set this value to 0.0001 in each experiment. In this scenario, we analyze the convergence rate expressed as the number of the iterations for the consensus - the number of the iterations averaged over all 1000 RGs⁵/100 RGGs⁶ are shown.

Thus, we test the chosen algorithms in 1000 unique random graphs for each p (i.e., in 4000 random graphs in overall), 100 unique densely, and 100 unique sparsely connected random geometric graphs. Therefore, each analyzed algorithm is tested in 4200 graphs in overall.

As mentioned above, we use two metrics (namely, MSE over the iterations and the convergence rate expressed as the number of the iterations for the consensus) to evaluate the performance of analyzed DLCDsAs. MSE is a reasonable metric used in various areas and is defined as follows [18]:

$$\text{MSE}(k) = \frac{1}{n} \cdot \sum_{i=1}^n \left(x_i(k) - \mathbf{1}^T \times \frac{\mathbf{x}(0)}{n} \right)^2 \quad (15)$$

The other used metric for performance evaluation is the convergence rate expressed as the number of the iterations for the consensus. The algorithms are considered to be completed at the first iteration when the condition from (14) is met.

⁴25th, 50th, 100th, 150th, 200th, 250th iteration are chosen

⁵for each p separately

⁶for each connectivity separately

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In the first experiment, we analyze selected DLCDsAs over RG formed by 50 vertices when the edge formation p takes these four values: 3.5%, 5%, 10%, and 15% (a higher probability of the edge formation p ensures higher connectivity of the generated graphs). From the results shown in Fig. 2, where MSE over the iterations is analyzed (we show the average over all 1000 graphs for each p), we can see that for $p = 3.5\%$ and at the 25th iteration (i.e., an early phase of the algorithms), the best performance is achieved by LD, the second one by MH, the third one by BC, the fourth one by MD, and OW achieves the worst performance. At the 50th iteration, OW (the fourth-best performing) outperforms MD (the worst algorithm in this phase). LD achieves lowest MSE at this iteration, MH is the second-best performing again, and BC achieves third lowest MSE. At the 100th iteration, OW outperforms also MH and BC and so is the second-best performing at this iteration - MD is the worst one, BC the second-worst, MH the third worst, and LD achieves the highest performance again. At the 150th iteration, BC slightly outperforms MH (the second-worst) and is thus the third worst. The order of the other algorithms is the same as at the previous analyzed iteration. At the 200th and the 250th iteration, OW takes smallest MSE among the examined algorithm and is thus the best performing in these phases. MD achieves the worst performance, MH the second-worst, BC the third one, and LD the fourth one. Over RGs with $p = 5\%$ at the 25th and the 50th iteration, the best performance is achieved by LD, the second one by OW, the third one by MH, the fourth one by BC, and highest MSE is taken by MD. At the 100th iteration, OW outperforms LD and is, therefore, the best performing (LD is the second one) - the order of the other algorithms is the same as at the previous analyzed iteration. At the 150th iteration, BC outperforms MH (the fourth-best performing) and is thus the third-best performing. OW achieves the highest performance again, LD is the second one, and MD takes highest MSE. This order is preserved also at the 200th and the 250th iteration. For $p = 10\%$, the best performance is achieved by

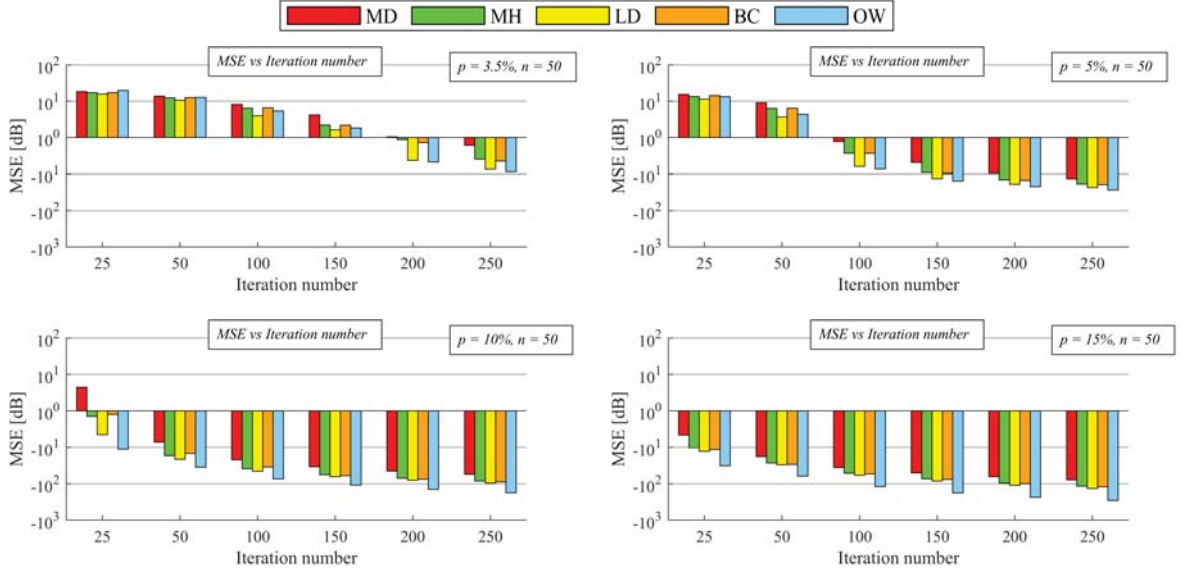


Fig. 2. MSE of MD, MH, LD, BC, and OW in decibels over iterations for edge formation $p = 3.5\%$, 5% , 10% , and 15% over random graphs (RGs) formed by 50 vertices ($n = 50$)

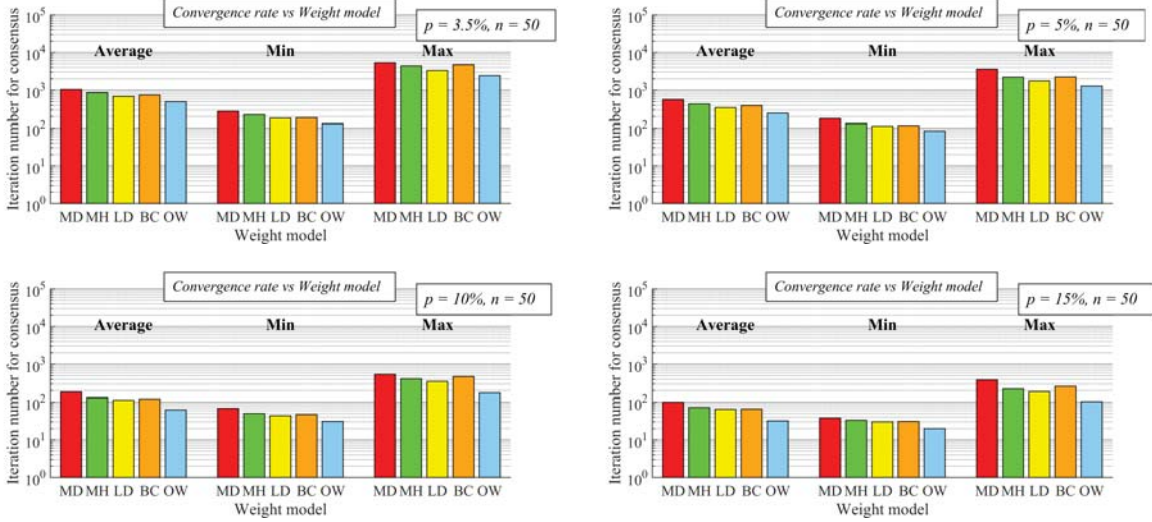


Fig. 3. Convergence rate (expressed as number of iterations for consensus) of MD, MH, LD, BC, and OW for edge formation $p = 3.5\%$, 5% , 10% , and 15% over random graphs (RGs) formed by 50 vertices ($n = 50$) - the algorithms are bounded by the stopping criterion provided in (14)

OW over the whole examined iteration interval, the second-best performing is LD, the third one is MH in earlier phases (i.e., at the 25th, 50th, 100th - here, BC is the fourth-best performing) and, BC in later phases (i.e., at the 150th, the 200th, the 250th iteration - here, MH is the fourth-best performing). MD takes highest MSE at each examined iteration. For $p = 15\%$ and at each examined iteration, OW achieves the highest performance, LD the second highest one, BC the third one, MH the fourth one, and MD is the worst again.

In this paragraph, we compare the selected algorithms over RGs in terms of the impact of the graph connectivity on MSE [dB] (see Fig. 3). In order to ensure easy readability of the text, we focus only on the 250th iteration, and the difference between $p = 3.5\%$ and $p = 15\%$. Here, we can see that OW is the most significantly affected by the graph topology as the difference in its MSE is equal to approx. 278 dB.

Then, LD is the second most affected (approx. 128 dB), BC the third one (approx. 117 dB), MH the fourth one (approx. 113 dB), and MD is the least affected among the examined algorithms (approx. 76 dB).

In Fig. 3, we show the results when DLCDsAs are bounded by the stopping criterion from (14). We can see that the highest average convergence rate is achieved by OW for each p , LD is the second fastest, BC is the third one, MH is the fourth one, and the worst performance is observable for MD. Furthermore, we analyze the maximum and the minimum from the convergence rates for each examined algorithm and for each p (i.e., we show the convergence rate in the graph where an algorithm is the slowest/the fastest). When the minimum is analyzed for each p , the order of the algorithm is the same as when the average is examined. However, when the maximum is analyzed, BC is outperformed by MH for each p .

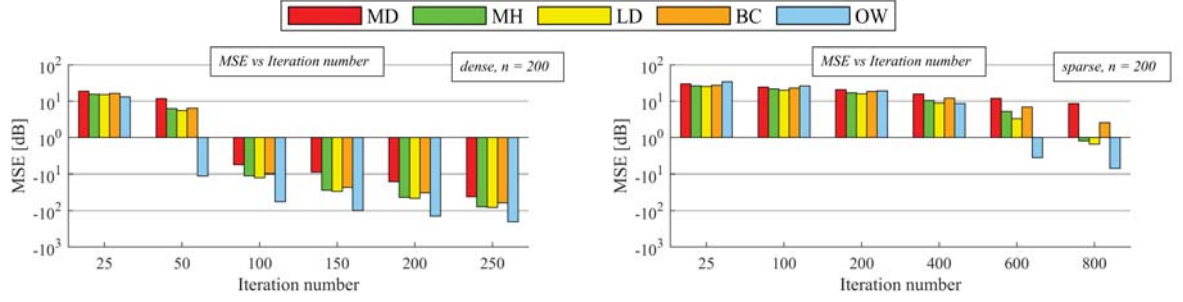


Fig. 4. MSE of MD, MH, LD, BC, and OW in decibels over iterations over random geometric graphs (RGGs) of dense and sparse connectivity formed by 200 vertices ($n = 200$)

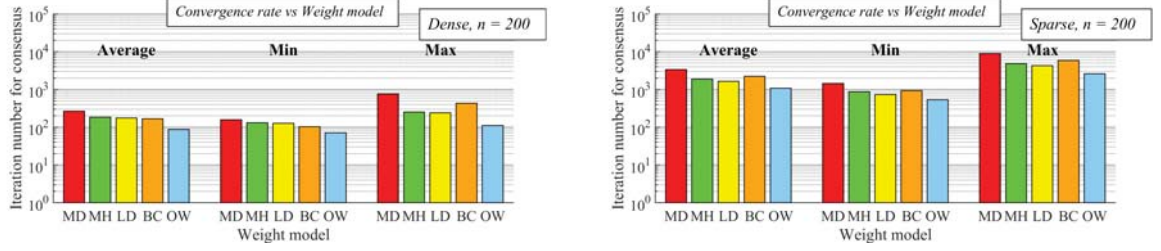


Fig. 5. Convergence rate (expressed as number of iterations for consensus) of MD, MH, LD, BC, and OW over random geometric graphs (RGGs) of dense and sparse connectivity formed by 200 vertices ($n = 200$) - the algorithms are bounded by the stopping criterion provided in (14)

In the next analysis, we examine MSE over the iterations of selected DLCDsAs over 100 RGGs of either dense or sparse connectivity. In densely connected RGGs (Fig. 4), we can see that OW outperform the concurrent algorithms over the whole examined interval. The second most precise is LD, MH is the third one, BC is the fourth one, and MD achieves the worst performance at each iteration. In sparsely connected RGGs (the algorithms are analyzed up to the 800th iteration compared to the previous analyses - it is because the algorithms achieve low precision even at the 250th iteration), we can see that OW is the worst performing in an earlier phase (i.e., at the 25th and the 100th iteration), the second-worst performing at the 200th iteration, and from the 400th iteration, it outperforms all the concurrent algorithms and so is the best performing at these iterations. The order of the other algorithms is as follows: the worst performance is achieved by MD (except on the 25th and the 100th iteration - when it is the second-worst), the second-worst by BC (except on the 25th, the 100th, and the 200th iteration - when it is the third worst), the third worst one by MH (except on the 25th, the 100th, and the 200th iteration - when it is the fourth-worst), and the fourth-worst one by LD (except on the 25th, the 100th, and the 200th iteration - when it is the best performing).

Now, like in RGs, we compare the analyzed algorithms over RGGs in terms of the impact of the graph connectivity on MSE [dB] (see Fig. 5). Again, we deal only with MSE at the 250th iteration. Here, we can see that OW is the most significantly affected by the graph topology again, and the difference in its MSE is equal to approx. 185 dB. The second most affected algorithm is LD (approx. 67 dB), MH is the third one (approx. 63 dB), BC is the fourth one (approx. 44 dB), and the

difference of MD takes only approx. 22 dB.

In the scenario when the algorithms are bounded by (14) over RGGs (see Fig. 5), we can see that in the densely connected graphs, OW has the highest average convergence rate among the examined DLCDsAs, BC is the second fastest, LD is the third one, MH the fourth one, and MD achieves the slowest convergence rate. In the sparsely connected graphs, OW has the fastest average convergence again, however, BC is outperformed by LD (the second fastest) and MH (the third fastest) and is, therefore, the fourth fastest. The slowest algorithm is MD again just like in the densely connected RGGs. Afterward, we show the maximum and the minimum for each algorithm in both densely and sparsely RGGs (i.e., we show the convergence rate in the graph where an algorithm is the slowest/the fastest). In the densely connected graphs, it can be seen that the order of the convergence rates corresponds to the order when the average over all 100 graphs is shown except on the maximum of BC - BC is outperformed by MH and LD even though it is better in terms of the average and the minimum. In the sparsely connected graphs, the order of the examined algorithms is the same when the average, the minimum, and the maximum are analyzed.

V. CONCLUSION

In this paper, we analyze five frequently applied deterministic linear consensus-based algorithms for distributed summing (namely, MD, MH, LD, BC, and OW) over random graphs with a varying edge formation and random geometric graphs of dense or sparse connectivity. From the experimentally obtained results, we can see that OW achieves lowest MSE over the iterations over all RGs except on in earlier phases over

graphs with lower connectivity (i.e., $p = 3.5\%$ and 5%). An increase in connectivity (ensured by increasing p) causes that OW outperforms the concurrent algorithms earlier. BC is also more performing in later phases. In earlier phases over the graphs with lower connectivity (i.e. $p = 3.5\%$ and 5%), LD achieves the highest performance, and this algorithm is very precise also in other graphs and phases. The worst performance is achieved by MD at almost every iteration and in each set of the graphs. Only except for the 25th iteration in the graphs with $p = 3.5\%$, this algorithm outperforms OW, significantly best performing algorithm in general. In the random geometric graphs, OW is the best performing over the whole examined interval in the densely connected graphs. In the sparsely connected ones, this algorithm achieves lowest MSE at later iterations (from the 400th iteration), however, in earlier phases (i.e., the 25th, the 100th iteration), it is outperformed even by MD, which achieves the worst performance again. In the other analyses, we compare selected DLCDSAs when the algorithm execution is bounded. From the results, we can see that OW outperforms the concurrent algorithms in all RGs and RGGs (this algorithm is also the most affected by the graph connectivities). The worst performance is achieved by MD in all RGs and RGGs (this algorithm is the least affected by the graph connectivities).

So, the final conclusion is that OW achieves the highest performance, and MD the lowest performance in both scenarios, i.e., the algorithms are/are not bounded by the implemented stopping criterion.

REFERENCES

- [1] D. Izadi, J. H. Abawajy, S. Ghanavati, and T. Herawan, "A data fusion method in wireless sensor networks," *Sensors*, vol. 15, no. 2, pp. 2964–2979, January 2015.
- [2] R. Mishra, V. Jha, R. K. Tripathi, and A. K. Sharma, "Design of Probability Density Function Targeting Energy Efficient Network for Coalition Based WSNs," *Wireless Pers. Commun.*, vol. 99, no. 2, pp. 651–680, March 2018.
- [3] N. Zhao, X. Chi, L. Zhao, and Y. Zhu, "A Spectrum Efficient Self-Admission Framework for Coexisting IEEE 802.15. 4 Networks under Heterogeneous Traffics," *Radioengineering*, vol. 27, no. 1, pp. 326–334, April 2018.
- [4] A. Kumar, H. Y. Shwe, K. J. Wong, and P. H. Chong, "A survey of network lifetime maximization techniques in wireless sensor networks," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 2, pp. 25–72, January 2017.
- [5] G. Han, L. Liu, J. Jiang, L. Shu, and G. Hancke, "Analysis of energy-efficient connected target coverage algorithms for industrial wireless sensor networks," *IEEE Commun. Surv. Tutor.*, vol. 13, no. 1, pp. 135–143, December 2017.
- [6] A. Tahbaz-Salehi and A. Jadbabaie, "A one-parameter family of distributed consensus algorithms with boundary: From shortest paths to mean hitting times," in *Proc. of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, 2006, pp. 4664–4669, December 2006.
- [7] J. Gutierrez-Gutierrez and M. Zarraga-Rodriguez, "Analysis of known linear distributed average consensus algorithms on cycles and paths," *Sensors*, vol. 18, no. 4, pp. 2964–2979, April 2018.
- [8] V. Schwarz and G. Matz, "Average consensus in wireless sensor networks: Will it blend?," in *Proc. IEEE SPAWC-2012*, Vancouver, BC, Canada, 2013, pp. 4584–4588, May 2013.
- [9] D. Merezeanu and M. Nicolae, "Consensus control of discrete-time multi-agent systems," *Politehn. Univ. Bucharest Sci. Bull. Ser. A Appl. Math. Phys.*, vol. 79, no. 1, pp. 167–174, 2017.
- [10] G. Stamatescu, I. Stamatescu, and D. Popescu, "Consensus-based Data Aggregation for Wireless Sensor Networks," *Control. Eng. Appl. Inf.*, vol. 19, no. 2, pp. 4669–4682, June 2017.
- [11] M. Kenyeres, J. Kenyeres, V. Skorpil, and R. Burget, "Distributed aggregate function estimation by Biphasically configured Metropolis-Hasting weight model," *Radioengineering*, vol. 26, no. 2, pp. 479–495, June 2017.
- [12] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, September 2004.
- [13] M. Kenyeres, J. Kenyeres, and I. Budinska, "Distributed linear summing in wireless sensor networks," in *Proc. of SAMI*, Herlany, Slovakia, 2019, pp. 23–30, Jan. 2019.
- [14] S. V. Macua et al. "How to implement doubly-stochastic matrices for consensus-based distributed algorithms," in *Proc. of SAM*, A Coruña, Spain, 2014, pp. 333–336, June 2014.
- [15] F. Haddadpour and V. R. Calambe, "Codes for distributed finite alphabet matrix-vector multiplication," in *Proc. of ISIT*, Vail, CO, USA, 2018, pp. 1625–1629, June 2018.
- [16] C. Bordenave, P. Caputo, D. Chafaï, and K. Tikhomirov, "On the spectral radius of a random matrix: An upper bound without fourth moment," *Ann. Probab.*, vol. 46, no. 4, 2018, pp. 2268–2286, July 2018.
- [17] A. Frieze and M. Karonski, *Introduction to random graphs*, Cambridge: Cambridge university press, 2016, p. 3–18, 212–220.
- [18] S. S. Pereira and A. Pages-Zamora, "Mean square convergence of consensus algorithms in random WSNs," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2866–2874, Sept. 2010.